Amendments to the Specification

On page 23, after lines 20, please insert the following paragraphs:

-Let MAP and PAM be two tables or functions. MAP maps an edge of the control-flow graph to a pin of the circuit. PAM maps a pin of the circuit to an edge of the control-flow graph. PAM is the inverse of MAP and MAP is the inverse of PAM, i.e. for any A, C, MAP(PAM(C)) = C and PAM(MAP(A)) = A.

FLOP is a table that maps state nodes of the control flow graph to flip-flops. For each state node there is exactly one flip-flop in FLOP; these flip-flops are unique, i.e. FLOP has as many entries as there are states in the control-flow graph.

The procedure CCT takes as its sole argument an arc C of the control-flow graph. CCT performs a recursive backward traversal of the control flow graph that builds the desired one-hot FSM as it traverses the control-flow graph, using only local information about the graph and the tables MAP, PAM, and FLOP.

CCT must be embedded in a loop that calls CCT once for each edge of the control-flow graph; this assures that all parts of the graph will be traversed.

A pseudocode description of the procedure CCT is shown here.

```
T = MAP(C);
if (T does not exist) {
  create a new output pin T;
  add the pair (C, T) to MAP and the pair (T, C) to PAM;
  N = { the node at the feather end of C }
  if (N is a state node) {
    Connect T to the Q pin of FLOP(N).
  }else if (N is a join node with K in-arcs) {
```

```
Let G be a new K-input OR gate;
Connect the output of G to T;
for (each in-arc A of N) {
  Let P = MAP(A);
  if (P does not exist) {
    create a new output pin P;
    add the pair (A, P) to MAP and the pair (P, A) to PAM;
    call CCT(A);
  }
  Connect P to an unconnected input pin of G.
}
}else if (N is a fork node) {
  Let G be a new 2-input AND gate;
  Connect the output of G to T;
  Let A be the in-arc of N.
  Let P = MAP(A).
  if (P does not exist) {
    create a new output pin P;
    add the pair (A, P) to MAP and the pair (P, A) to PAM;
    call CCT(A);
  }
  Connect one input of G to P;
  Make the other input of G a primary status input corresponding
    to the branch condition that is annotated onto A.
```

```
  }else if (N is the reset node) {
    Connect T to the reset input.
  }else{
    Let A be the in-arc of N.
    Let P = MAP(A).
    if (P does not exist) {
      create a new output pin P;
      add the pair (A, P) to MAP and the pair (P, A) to PAM;
      call CCT(A);
    }
    Connect P to T.
  }
}else{
  /* Nothing need be done; C has been traversed already. */
}.-
```